1. A method for use in a distributed system for processing a knowbot program that has the ability to move from node to node in the distributed system comprising

in an operating environment in each of the nodes, providing service facilities useful to the knowbot program, and

in the operating environment running a supervisor process that enables the knowbot program to make use of the service facilities but does not permit direct access by the knowbot program to facilities of the operating environment.

2. The method of claim 1 further comprising creating a bastion object in the unrestricted environment to protect the unrestricted environment and passing it into a restricted environment within which the knowbot program is running.

3. The method of claim 2 in which the bastion object provides an interface for the knowbot program to access the service facilities in a safe manner and which is substantially the same interface as the interface that the service facilities provide in the unrestricted environment.

4. The method of claim 2 in which the bastion object performs type checking on all method calls made by a knowbot program to a service facility.

5. A method for use in a distributed system for processing a knowbot program that executes in one node of the distributed system, may be interrupted at almost any point in its execution, and may be moved to another node of the distributed system for further execution, comprising

in the one node, capturing a current state of the knowbot program execution,

delivering the captured state and program code of the knowbot program to the other node, and

continuing execution at the other node from the point of interruption based on the captured state and the program code.

6. The method of claim 5 further comprising

also delivering with the captured state and the program code a transported file system or other information created during execution of the knowbot program.

7. The method of claim 6 in which the information in the transported file system or other information is accessible without executing the knowbot program.

8. The method of claim 5 in which the step of capturing comprises using an encoding scheme of a language interpreter.

9. A method for enabling communication with a knowbot program running in a distributed system, a knowbot service station, an extension, or another application, comprising

providing a connector mechanism which permits each of the knowbot programs, knowbot service stations, extensions, and other applications to identify services that it provides, and permits each of them to find services that it needs, and

enabling knowbot programs to communicate with knowbot service stations via connector objects associated with the connector mechanism.

10. The method of claim 9 in which the connector object is provided by a supervisor process running in the distributed environment and the connector object prevents uncontrolled access to a needed service.

11. The method of claim 9 in which the connector mechanism includes a connector broker and connector manager.

12. The method of claim 9 in which the connector objects are data typed.

13. A method for enabling negotiation between two unrelated knowbot programs, knowbot service stations, extensions, or other applications, in a distributed system, comprising

in an operating environment in a node of the distributed system, receiving information from one of the two knowbot programs, knowbot service stations, extensions, or other applications, concerning a transaction offered to other knowbot programs, knowbot service stations, extensions, or other applications,

in the operating environment in the node, receiving information from the other of the two knowbot programs, knowbot service stations, extensions, or other applications concerning a transaction in which the other of the knowbot programs, knowbot service stations, extensions, and other applications wishes to engage,

notifying the other knowbot program, knowbot service station, extension, or other application of the one knowbot program, knowbot service station, extension, or other application, and

enabling the two knowbot programs, knowbot services stations, extensions, or other applications to communicate concerning the transaction.

14. The method of claim 13 in which the information is received from the two knowbot programs by a third knowbot program.

15. A method for enabling action by an operating environment in a distributed system with respect a knowbot program which is programmed in a language that is not fully supported by the operating environment, comprising

labeling a knowbot program to identify operating environment features required for full support of the knowbot program,

in an operating environment, examining the labeling of the knowbot program to determine whether the operating environment supports all of the identified features, and

taking an action based on whether all the identified features are supported.

16. The method of claim 15 wherein the action comprises sending the knowbot program to another operating environment for processing.

17. The method of claim 15 in which the action comprises retrieving non-program specific data from the knowbot program.

18. A method for aiding communication with a knowbot program executing in operating environments provided at nodes of the distributed system, comprising

maintaining a name space that uniquely identifies types of information to be interchanged, and

using a name within the name space to identify the type of information to be interchanged.

19. The method of claim 18 in which the knowbot program registers an interface which includes the name of a type of information that is to be interchanged.

20. A method for controlling the timing of execution of an action associated with a knowbot program running in an operating environment provided at a node of a distributed system, comprising

providing a trigger protocol in the operating environment,

enabling the knowbot program to register a condition with the operating environment,

causing the operating environment to trigger the execution of the action upon the occurrence of the condition.

21. The method of claim 20 in which the trigger protocol defines trigger statements each of which identifies at least the condition and the action.

22. The method of claim 20 in which the operating environment maintains a table of registered trigger expressions for all knowbot programs that have registered conditions.

23. The method of claim 20 in which the execution is triggered by a program contained in the knowbot program.

24. A method for controlling interaction between a knowbot program and an application running in an operating environment provided at a node of a distributed system, comprising

defining a trusted portion of the operating environment which provides trusted services to the knowbot program,

requiring portions of the application running in the operating environment to be registered as trusted, and

permitting indirect interaction via the operating environment between the knowbot program and the application running in the operating environment only if the portions of the application required to be registered have been registered.

25. A method for enabling a knowbot program to carry out defined functions including otherwise unsafe functions, thorugh the use of extensions comprising

coding safe extensions to an operating environment and to the interpretive language under which the knowbot program runs, and

permitting the knowbot program to carry out the defined functions by making use of the extensions.